Computing the Mean of Geometric Features Application to the Mean Rotation

Xavier Pennec

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Computing the mean of geometric features Application to the mean rotation

Xavier Pennec

N˚ 3371

March 1998

THÈME 3

ISSN 0249-6399

INRIA
SOPHIA ANTIPOLIS

Computing the mean of geometric features Application to the mean rotation

Xavier Pennec*

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet Epidaure

Rapport de recherche n° 3371 — March 1998 — 16 pages

Abstract: The question we investigate in this article is: what is the mean value of a set of geometric features and how can we compute it? We use as a guiding example one of the most studied type of features in computer vision and robotics: 3D rotations.

The usual techniques on points consist of minimizing the least-square criterion, which gives the barycenter, the weighted least-squares or the sum of (squared) Mahalanobis distances. Unfortunately, these techniques rely on the vector space structure of points and generalizing them directly to other types of features could lead to paradoxes [PA97]. For instance, computing the barycenter of rotations using rotation matrices, unit quaternions or rotation vectors gives three different results.

We present in this article a thorough generalization of the three above criterions to homogeneous Riemannian manifolds that rely only on intrinsic characteristics of the manifold. The necessary condition for the mean rotation, independently derived in [Den96], is obtained here as a particular case of a general formula. We also propose an intrinsic gradient descent algorithm to obtain the minimum of the criterions and show how to estimate the uncertainty of the resulting estimation. These algorithms prove to be not only accurate but also efficient: computations are only 3 to 4 times longer for rotations than for points. The accuracy prediction of the results is within 1%, which is quite remarkable.

The striking similarity of the algorithms' behavior for general features and for points stresses the validity of our approach using Riemannian geometry and lets us anticipate that other statistical results and algorithms could be generalized to manifolds in this framework.

Key-words: mean feature, mean rotation, fusion, Riemannian geometry

* INRIA, projet EPIDAURE, and MIT, Artificial Intelligence Lab

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, B.P. 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : 04 92 38 77 77 - International : +33 4 92 38 77 77 — Fax : 04 92 38 77 65 - International : +33 4 92 38 77 65

Calculer la moyenne de primitives géométriques Application au cas des rotations

Résumé : Nous nous intéressons dans cet article à la question suivante : qu'est-ce que la moyenne d'un ensemble de primitives géométriques et comment la calculer ? Nous prennons comme illustration l'un des types de primitives le plus utilisé en vision par ordinateur et en robotique: les rotations 3D.

Les techniques usuelles pour les points consitent à minimiser le critère des moindres carrés, ce qui donne le barycentre, les moindres carrés pondérés ou la somme des distances de Mahalanobis (au carré). Malheureusement, ces techniques reposent sur la structure d'espace vectoriel des points et nous avons déjà montré que leur généralisation hative à d'autres types de primitives conduit à des paradoxes. Par exemple, le calcul du barycentre de rotations en utilisant les matrices de rotation, les quaternions unitaires ou le vecteur rotation donne trois résultats différents.

Nous présentons dans cet article une généralisation rigoureuse au cas de variétés riemanniennes homogènes des trois critères évoqués ci-dessus. Cette generalisation repose sur caractéristiques intrinsèques de la variété considérée. Les conditions nécéssaires caractérisant la rotation moyenne, dérivées indépendemment dans [Den96], sont ici obtenues comme un cas particulier de la formule générale. Nous proposons également un algorithme de descente de gradient intrinsèque à la variété pour obtenir le minimum des critères et nous montrons comment estimer l'incertitude sur notre estimation résultante. Ces algorithmes apparaissent non seulement précis mais aussi efficaces pusisqu'ils ne prennent que 3 à 4 fois plus de temps pour les rotations que pour les points. La prédiction d'incertitude est correcte à moins de 1% près, ce qui est également remarquable.

La similarité frappante entre le comportement de ces algorithmes pour des primitives générales et pour les points souligne la validité de notre approche théorique utilisant la géométrie riemannienne et nous permet d'anticiper la généralisation d'autres résultats et algorithmes statistiques aux variétés en utilisant le même cadre

Mots-clés : primitive moyenne, rotation moyenne, fusion, geometrie riemannienne

1 Introduction

Assume that we have a set of measures $\{x_i\}$ of the same point x and that we want to obtain a better estimate. The usual way is to choose the barycenter $\bar{x} = \frac{1}{n}\sum_i x_i$ which is the least-squares estimation of the mean value. Now, if we have a confidence value p_i on each measure, we compute the weighted barycenter, which minimizes the weighted least-squares criterion $C(x) = \sum_i p_i.\|x_i - x\|^2$. If we have a covariance matrix $\Sigma_{\boldsymbol{x_i x_i}}$ describing the uncertainty of each measure, then we have to minimize the sum of squares Mahalanobis distances: $C(x) = \sum_i (x_i - x)^{\mathrm{T}}.\Sigma_{\boldsymbol{x_i x_i}}(x_i - x)$.

All these techniques are well known and are easy to implement. However, geometric models of the real world often lead to consider more complex features like lines [Gri90], planes, oriented points [FA96] or frames [PT97]. The problem we tackle in this article is how to generalize the previous methods for generic geometric features. This problem is not so easy: we have shown in [PA97] that geometric features generally do not belong to a vector space but rather to a manifold and that it induces paradoxes if we try to use the standard techniques on points with them. For instance, we can represent a 3-D rotation by its matrix R, the corresponding unit quaternions $\pm q$ or the rotation vector $r = \theta.n$. Using the barycenter to compute the mean, we obtain either $\underline{R} = \frac{1}{n}\sum_i R_i$, $\underline{q} = \frac{1}{n}\sum_i q_i$ or $\underline{r} = \frac{1}{n}\sum_i r_i$. The three results correspond to different rotations[1].

From a mathematical point of view, all operations we define on features should rely on intrinsic characteristics of the manifold and not on the vector properties of some particular chart. Moreover, there is generally a transformation group acting on the manifold that models the possible image view points and/or the subject movement in the images. Any operation should be invariant or "covariant" with respect to the action of this group. For instance, the barycenter of points is invariant under the action of rigid or affine transformations: $A(\sum x_i) = \sum Ax_i$. In the case of geometric features, these desiderata are much more difficult to meet, and we have to take a look at quite deep results in differential geometry in order to have an insight of what are the intrinsic characteristics of the manifold we can use. This is the object of section 2 which summarizes the theory developed in [Pen96]. In section 3, we investigate the least-squares minimization of the Riemannian distance and we show that it can be solved efficiently. Then, we generalize this algorithm to the weighted least-squares and the Mahalanobis criterions in section 4. In the last section, we analyze and compare the three algorithms in terms of accuracy of the estimated mean, accuracy of their uncertainty estimation and computation time.

2 Geometric features

We review in this section some basic and deeper notions of Riemannian geometry. This synthesis was inspired by [Spi79, chap. 9], [Kli82] and [Car92].

2.1 Riemannian manifolds

In the geometric framework, one specifies the structure of a manifold $\mathcal{M}$ by a *Riemannian metric*. This is a continuous collection of dot products on the tangent space at each point

[1] The first two are not even rotations unless they are renormalized: the sum of orthogonal matrices is generally not an orthogonal matrix and the sum of unit quaternions is not a unit quaternion. Moreover, choosing the sign of the unit quaternion used in the sum is not a trivial problem.

x of the manifold. Thus, if we consider a curve on the manifold, we can compute at each point its instantaneous speed vector and its norm, the instantaneous speed. To compute the length of the curve, we can proceed as usual by integrating this value along the curve. The distance between two points of a connected Riemannian manifold is the minimum length among the curves joining these points. The curves realizing this minimum for any two points of the manifold are called geodesics. The calculus of variations shows that geodesics are the solutions of a system of second order differential equations depending on the Riemannian metric.

In this article, we assume that the manifold is *geodesically complete*, i.e. that the definition domain of all geodesics can be extended to $\mathbb{R}$. This means that the manifold has no boundary nor any singular point that we can reach in a finite time. As an important consequence, the Hopf-Rinow-De Rham theorem states that there always exist at least one minimizing geodesic between any two points of the manifold (i.e. which length is the distance between the two points).

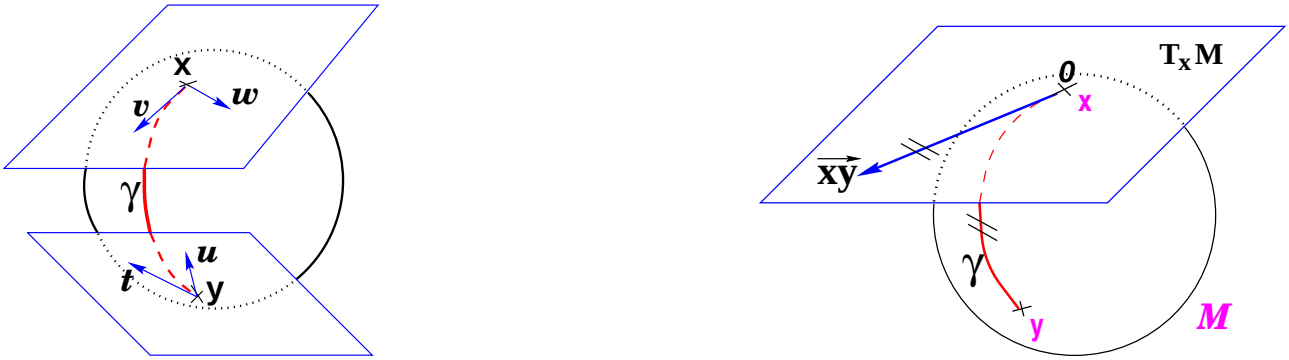

Figure 1: **Left:** The tangent planes at points x and y of the sphere $\mathcal{S}_2$ are different: the vectors v and w of $T_{\mathrm{x}}\mathcal{M}$ cannot be compared to the vectors t and u of $T_{\mathrm{y}}\mathcal{M}$. Thus, it is natural to define the dot product on each tangent plane. **Right:** The geodesics starting at x are straight lines in the exponential map and the distance along them is conserved.

Exponential chart From the theory of second order differential equations, we know that there exits one and only one geodesic starting at a given feature x with a given tangent vector. This allows to develop the manifold in the tangent space along the geodesics (think of rolling a sphere along its tangent plane at a given point). The geodesics going through this point are transformed into straight lines and the distance along these geodesics are conserved (at least in a neighborhood of x). The function that maps to each vector the corresponding point on the manifold is called the *exponential map*.

This map is defined in the whole tangent space $T_{\mathrm{x}}\mathcal{M}$ (since the manifold is geodesically complete) but it is one-to-one only locally around the origin. If we look for the maximal domain where this map is one-to-one, we find out that it is a star-shaped domain delimited by a continuous curve C_{x} called the *tangential cut-locus*. The image of C_{x} by the exponential map is the cut locus $\mathcal{C}_{\mathrm{x}}$ of point x. Roughly, this is the set of points where several minimizing geodesics starting from x meet[2]. On the sphere $\mathcal{S}_2(1)$ for instance, the cut locus of a point x is its antipodal point and the tangential cut locus is the circle of radius π.

The exponential map within this domain realizes a chart called *the exponential chart*. It covers all the manifold except the cut locus of the development point, which has a null measure.

[2]More precisely, the cut locus is the closure of this set.

Let $\overrightarrow{\mathrm{xy}}$ be the representation of y in this chart. Then its distance to the origin is $\mathrm{dist}(\mathrm{x},\mathrm{y}) = \|\vec{\mathrm{y}}\|_{\mathrm{x}}$. This chart is somehow the "most linear" chart of the manifold with respect to the feature x.

2.2 Homogeneous manifolds

Invariant metric Now, since we are working with features on which acts a transformation group that models the possible image view points, it is natural to choose an invariant Riemannian metric (the existence conditions are detailed in [Pen96]). This way, all the measurements based on distance are independent of the image reference frame or the transformation of the image: $\mathrm{dist}(\mathrm{x},\mathrm{y}) = \mathrm{dist}(\mathrm{g}\star\mathrm{x},\mathrm{g}\star\mathrm{y})$.

Let o be a point of the manifold that we call the origin: we call *principal chart* the exponential chart at the origin for the invariant metric and we denote by $\vec{\mathrm{x}}$ the representation of x in this chart. Assuming that we have chosen an ortho-normal coordinate system, the distance with the origin is: $\mathrm{dist}(o,\mathrm{x}) = \|\vec{\mathrm{x}}\|$. Now, let $\mathrm{f_y}$ be a "placement function", i.e. a transformation such that $\mathrm{f_y}\star o = \mathrm{y}$. The distance between any two points is: $\mathrm{dist}(\mathrm{x},\mathrm{y}) = \mathrm{dist}\left(\mathrm{f_y^{(-1)}}\star\mathrm{x}, o\right) = \|\mathrm{f_{\vec{y}}^{(-1)}}\star\vec{\mathrm{x}}\|$. One can verify that this formula is independent of the choice of the placement function.

Exponential chart and exponential map at other points The previous formula shows that the vector $(\mathrm{f_{\vec{y}}^{(-1)}}\star\vec{\mathrm{x}})$ is the representation of x in an exponential chart at feature y. The coordinate system of this chart is ortho-normal but it depends on the chosen placement function. It is often more interesting to use the non-orthogonal coordinate system induced by the principal chart, which is independent of the placement function: this is the vector $\overrightarrow{\mathrm{yx}} = J(\mathrm{f_{\vec{y}}}).(\mathrm{f_{\vec{y}}^{(-1)}}\star x)$, where $J(\mathrm{f_{\vec{x}}}) = \left.\frac{\partial(\vec{\mathrm{f}}\star\vec{\mathrm{x}})}{\partial\vec{\mathrm{x}}}\right|_{\vec{\mathrm{x}}=o}$ is the Jacobian of the translation of the origin in the principal chart.

From a practical point of view, this means that we can "translate" the local calculations on points to the principal chart of our manifold by replacing $\overrightarrow{yx} = x - y$ with $\overrightarrow{\mathrm{yx}}$ and $x + \overrightarrow{\delta\mathrm{x}}$ with $\exp_{\vec{\mathrm{x}}}(\overrightarrow{\delta\mathrm{x}}) = \mathrm{f_{\vec{x}}}\star(J(\mathrm{f_{\vec{x}}})^{(-1)}.\vec{\mathrm{v}})$. For instance, the empirical covariance matrix with respect to a point y, becomes [Pen96]:

$$\Sigma_{\vec{\mathbf{x}}\vec{\mathbf{x}}}(\vec{\mathrm{y}}) = \frac{1}{n}\sum_{i=1}^{n}\overrightarrow{\mathrm{yx_i}}.\overrightarrow{\mathrm{yx_i}}^{\mathrm{T}} = \frac{1}{n}J(\mathrm{f_{\vec{y}}}).\left(\sum_{i=1}^{n}\left(\mathrm{f_{\vec{y}}^{(-1)}}\star\vec{\mathrm{x}}_i\right).\left(\mathrm{f_{\vec{y}}^{(-1)}}\star\vec{\mathrm{x}}_i\right)^{\mathrm{T}}\right).J(\mathrm{f_{\vec{y}}})^{\mathrm{T}} \qquad (1)$$

Atomic operations In fact, we have shown that most of the interesting operations on deterministic and uncertain geometric features can be expressed in the principal chart of the manifold with only the few operations we have introduced. From a computational point of view, this means that the only operations we have to implement for each type of feature are **the action of a transformation** $(\mathrm{f}\star\vec{\mathrm{x}})$ with its Jacobians $\frac{\partial(\mathrm{f}\star\vec{\mathrm{x}})}{\partial\mathrm{f}}$ and $\frac{\partial(\mathrm{f}\star\vec{\mathrm{f}})}{\partial\vec{\mathrm{x}}}$, and **the placement function** $\mathrm{f_{\vec{x}}}$ with its Jacobian $\frac{\partial(\mathrm{f_{\vec{x}}})}{\partial\vec{\mathrm{x}}}$. Every higher level operation can be expressed as a combination of these ones and thus can be implemented independently of the considered type of feature. To simplify further computations, we can add to these atomic operations the Jacobian of the the translation of the origin $J(\mathrm{f_{\vec{x}}})$.

2.3 Example on rotations

The case of 3D rotations is interesting as it forms not only a manifold but also the transformation group $\mathcal{SO}_3$. Hence, this is implicitly an homogeneous manifold as the group acts on itself. An

invariant metric should ensure that $\text{dist}(R_1, R_2) = \text{dist}(R_2^{\mathrm{T}}.R_1,\ Id)$. Hence we can focus on $\text{dist}(R,\ Id)$. It is well known that R can be characterized by an angle θ around an axis n (unit vector), the angle being null for the identity.

It turns out that the invariant Riemannian distance on $\mathcal{SO}_3$ is $\text{dist}(R_1, R_2) = \theta(R_2^{\mathrm{T}}.R_1)$ (see [Alt86, Pen96]). The geodesics starting from the identity for this metric are $\gamma(t) = R(\theta_0.t, n)$ and the exponential map at identity is the rotation vector $r = \theta.n$. As we have $R(\pi, n) = R(-\pi, n)$, the cut locus of the identity is the set of reflection rotations. Hence, the definition domain of the principal chart is the open ball $\mathcal{B}^3(0, \pi)$. From a computational point of view, we use the closed ball to represent all rotations but we have to remember that symmetric points are identified at the boundary of the domain: $r = \pi.n \ \equiv \ r' = -\pi.n$.

Since the $\mathcal{SO}_3$ is a group, the action is the composition and the placement function is uniquely determined: this is the identity ($\mathrm{f}_r = r$). If $R(r)$ and $r(R)$ denote the mappings between rotation vectors and matrices, the action is $r_2 \star r_1 = r_2 \circ r_1 = r(R(r_2).R(r_1))$. However, we note that the derivatives can be more simply computed using the composition of unit quaternions (equations are given in [Pen96] and [PT97]).

3 Least-square Riemannian distances

The least-squares criterion is easily generalized to any metric space and its minimum corresponds to the empirical mean value in a vector space. This observation is the basis of the theory of expectation for random variables in metric spaces proposed by Fréchet (see [PA97]). In our case, we can go further thanks to the invariant distance and write it:

$$C(\mathrm{y}) = \frac{1}{2}.\sum_i \text{dist}(\mathrm{y}, \mathrm{x}_i)^2 = \frac{1}{2}.\sum_i \text{dist}(\mathrm{f}_{\vec{\mathrm{y}}}^{(-1)} \star \vec{\mathrm{x}}_i, o)^2$$

Error vector Now, thanks to the good properties of the principal chart, it turns out that this criterion can be expressed as a classical sum of square of vector norms. Let $\vec{\mathrm{z}}_i = \mathrm{f}_{\vec{\mathrm{y}}}^{(-1)} \star \vec{\mathrm{x}}_i$ be the "error vector" in the principal chart. The criterion becomes

$$C(\mathrm{y}) = \frac{1}{2}.\sum_i \|\vec{\mathrm{z}}_i\|^2$$

We note that this expression is only true in the principal chart.

To minimize the criterion and determine the uncertainty at the minimum, we will need to compute its first and second derivatives. From the atomic operations, and using the composition rule for differentials, we can compute the error vector $\vec{\mathrm{z}}_i$ and its Jacobians $\frac{\partial \vec{\mathrm{z}}_i}{\partial \vec{\mathrm{x}}_i} = \frac{\partial \mathrm{f}_{\vec{\mathrm{y}}}^{(-1)} \star \vec{\mathrm{x}}_i}{\partial \vec{\mathrm{x}}_i}$, and $\frac{\partial \vec{\mathrm{z}}_i}{\partial \vec{\mathrm{y}}} = \frac{\partial \mathrm{f}_{\vec{\mathrm{y}}}^{(-1)} \star \vec{\mathrm{x}}_i}{\partial \mathrm{f}_{\vec{\mathrm{y}}}^{(-1)}}.\frac{\partial \mathrm{f}_{\vec{\mathrm{y}}}^{(-1)}}{\partial \mathrm{f}_{\vec{\mathrm{y}}}}.\frac{\partial \mathrm{f}_{\vec{\mathrm{y}}}}{\partial \mathrm{y}}$, If the uncertainty of the measurements $\vec{\mathrm{x}}_i$ is represented by the covariance matrix $\Sigma_{\vec{\mathbf{x}}_\mathbf{i}\vec{\mathbf{x}}_\mathbf{i}}$, the uncertainty of the error vector is simply:

$$\Sigma_{\vec{\mathbf{z}}_\mathbf{i}\vec{\mathbf{z}}_\mathbf{i}} = \frac{\partial \vec{\mathrm{z}}_i}{\partial \vec{\mathrm{x}}_i}.\Sigma_{\vec{\mathbf{x}}_\mathbf{i}\vec{\mathbf{x}}_\mathbf{i}}.\frac{\partial \vec{\mathrm{z}}_i}{\partial \vec{\mathrm{x}}_i}^{\mathrm{T}} = \frac{\partial (\mathrm{f}_{\vec{\mathrm{y}}}^{(-1)} \star \vec{\mathrm{x}}_i)}{\partial \vec{\mathrm{x}}_i}.\Sigma_{\mathbf{x}_\mathbf{i}\mathbf{x}_\mathbf{i}}.\frac{\partial (\mathrm{f}_{\vec{\mathrm{y}}}^{(-1)} \star \vec{\mathrm{x}}_i)}{\partial \vec{\mathrm{x}}_i}^{\mathrm{T}}$$

Derivatives of the criterion Since the criterion is $C = \frac{1}{2}\sum_i \vec{\mathrm{z}}_i^{\mathrm{T}}.\vec{\mathrm{z}}_i$, the first derivative is:

$$\Phi = \left(\frac{\partial C}{\partial \vec{\mathrm{y}}}\right)^{\mathrm{T}} = \sum_i \frac{\partial \vec{\mathrm{z}}_i}{\partial \vec{\mathrm{y}}}^{\mathrm{T}}.\vec{\mathrm{z}}_i$$

Neglecting the term in $\ddot{z}.z$ with respect to the terms in $\dot{z}^2$ in the second derivatives (see [GMW81, section 4.7]), we obtain:

$$H = \frac{\partial \Phi}{\partial \vec{y}} \simeq \sum_i \frac{\partial \vec{z}_i}{\partial \vec{y}}^{\mathrm{T}} . \frac{\partial \vec{z}_i}{\partial \vec{y}} \qquad \text{and} \qquad \frac{\partial \Phi}{\partial \vec{z}_i} \simeq \frac{\partial \vec{z}_i}{\partial \vec{y}}^{\mathrm{T}}$$

In this particular case, we can exploit some properties of the invariant metric to obtain more simple formulas (details in [Pen96]). We have:

$$\Phi = -J(\vec{f}_{\vec{y}})^{(-\mathrm{T})} \sum_{i=1}^{n} \vec{z}_i \qquad \text{and} \qquad H = n.J(\vec{f}_{\vec{y}})^{(-\mathrm{T})}.J(\vec{f}_{\vec{y}})^{(-1)} \qquad \text{and} \qquad \frac{\partial \Phi}{\partial \vec{z}_i} = -J(\vec{f}_{\vec{y}})^{(-\mathrm{T})} \qquad (2)$$

3.1 Characterization of the mean feature

A necessary condition for $\vec{y}$ to be a minimum of the criterion is that the derivative Φ is null. In our case, this gives:

$$\sum_i \vec{z}_i = 0 = \sum_i (f_{\vec{y}}^{(-1)} \star \vec{x}_i)$$

It means that the mean value and the barycenter correspond *in the exponential chart at the mean value:* they are both null.

For rotations, this necessary condition can be written:

$$\sum_i \bar{r}^{(-1)} \circ r_i = o \qquad \text{or} \qquad \sum_i r(\bar{R}^{\mathrm{T}}.R_i) = 0$$

This condition was independently derived (for rotations only) in [Den96] using the Lagrangian formulation of the least-squares optimization problem. However, he uses an approximation of this necessary condition to obtain a solution. In contrast, we propose in the next section an intrinsic and generic gradient descent algorithm to find a solution of the exact criterion. In [ZRS92], reviewed optimal and sub-optimal algorithms to compute the minimum of other rotation fusion criterions. However, if these solutions are usually closed to our in the case of a small error, they can differ significantly for larger deviations. Moreover, they are limited to rotations and cannot be easily generalized to other types of features.

3.2 A gradient descent algorithm

Let $\vec{y}$ be an estimation of the mean feature. *In the principal chart*, we have the following Taylor expansion (remember that we have to replace $y + \overrightarrow{\delta y}$ by $\exp_{\vec{y}}(\overrightarrow{\delta y})$):

$$C\left(\exp_{\vec{y}}(\overrightarrow{\delta y})\right) = C(\vec{y}) + \Phi^{\mathrm{T}}.\overrightarrow{\delta y} + \frac{1}{2}\overrightarrow{\delta y}^{\mathrm{T}}.H.\overrightarrow{\delta y} + O\left(\|\overrightarrow{\delta y}\|^3\right) \qquad (3)$$

The minimum of this approximated criterion is given for:

$$\overrightarrow{\delta y} = -H^{(-1)}.\Phi$$

Reporting the values of equation (2), we find the very simple formula: $\overrightarrow{\delta y} = J(f_{\vec{y}}).\frac{1}{n}.\sum_{i=1}^{n} \vec{z}_i$. Hence, the minimum of the approximated criterion is realized for

$$\hat{\vec{y}} = \exp_{\vec{y}}(\overrightarrow{\delta y}) = f_{\vec{y}} \star \left(J(f_{\vec{y}})^{(-1)}.\overrightarrow{\delta y}\right) = f_{\vec{y}} \star \left(\frac{1}{n}.\sum_{i=1}^{n} \vec{z}_i\right)$$

Now, we can obtain a much better estimation of the transformation by iterating the previous algorithm. As an initial estimate, we can choose any feature $\vec{\mathrm{x}}_i$ if nothing else is given. Let $\vec{\mathrm{y}}_t$ be the estimation of the transformation at step t. The next estimation is

$$\vec{\mathrm{y}}_{t+1} = \exp_{\vec{\mathrm{y}}_t}(\overrightarrow{\delta \mathrm{y}}_t) = \vec{\mathrm{f}}_{\vec{\mathrm{y}}_t} \star \left(\frac{1}{n} \sum_{i=1}^{n} \left(\vec{\mathrm{f}}^{(-1)}_{\vec{\mathrm{y}}_t} \star \vec{\mathrm{x}}_i \right) \right) \tag{4}$$

The process is stopped when the norm $\|\overrightarrow{\delta \mathrm{y}}_t\|$ of the adjustment vector becomes too small (we use $\varepsilon = 10^{-10}$, which is almost the numeric precision of the machine) or when the number of iterations becomes too high. Practically, we observe that this algorithm always converges in about 5 iterations for any type of features. In the case of points, the equation is simplified to $y_{t+1} = \frac{1}{n} \sum_i x_i$, which gives the barycenter in only one iteration from any starting point.

3.3 Estimation of the uncertainty at the minimum

We have obtained above the transformation minimizing the criterion for the observed data. Now, since the data are corrupted by noise (represented by their covariance matrix), the estimated mean value could be slightly different from the exact one. We now quantify the change in the state vector with respect to the change in the data.

Let us assume for the moment that both the data and the state are vectors. Let $\hat{\chi}^{\mathrm{T}} = (\hat{z}_1^{\mathrm{T}}, \dots \hat{z}_n^{\mathrm{T}})$ be the vector of observed data (the vector of all error vectors) and $\hat{y}$ the corresponding mean value (we have added here the hats to stress the difference between actual and potential measurements). The minimum $y(\chi)$ of the criterion for a data vector χ is characterized by $\Phi = \left(\frac{\partial C}{\partial y}^{\mathrm{T}} \right) = 0$. Let $\hat{H}$ and $\hat{J}_\Phi = \frac{\partial \Phi}{\partial \chi \partial y}$ be the values of the second order derivatives of the criterion at the actual values $(\hat{\chi}, \hat{y})$. Since $\hat{y}$ minimizes the criterion for the data $\hat{\chi}$, we have $\Phi(\hat{\chi}, \hat{y}) = 0$ and a Taylor expansion around these values gives:

$$\Phi(\chi, y) = \hat{J}_\Phi.(\chi - \hat{\chi}) + \hat{H}.(y - \hat{y}) + O(\|y - \hat{y}\|^2) + O(\|\chi - \hat{\chi}\|^2)$$

Equating to zero, we obtain a second order approximation of relation between the state y

$$y - \hat{y} = -\hat{H}^{(-1)}.\hat{J}_\Phi.(\chi - \hat{\chi}) + O(\|\chi - \hat{\chi}\|^2)$$

Thus, the covariance of $\hat{y}$ is:

$$\Sigma_{\hat{\boldsymbol{y}}\hat{\boldsymbol{y}}} = \mathbf{E}\left((y - \hat{y}).(y - \hat{y})^{\mathrm{T}}\right) = \hat{H}^{(-1)}.\hat{J}_\Phi.\Sigma_{\hat{\boldsymbol{\chi}}\hat{\boldsymbol{\chi}}}\hat{J}_\Phi^{\mathrm{T}}.\hat{H}^{(-1)}$$

Assuming that all our measurements are independent, we have $\Sigma_{\boldsymbol{\chi}\boldsymbol{\chi}} = \mathrm{DIAG}(\Sigma_{\boldsymbol{z_1 z_1}}, \dots \Sigma_{\boldsymbol{z_n z_n}})$ and we can simplify $\hat{J}_\Phi.\Sigma_{\hat{\boldsymbol{\chi}}\hat{\boldsymbol{\chi}}}\hat{J}_\Phi^{\mathrm{T}}$ to obtain:

$$\Sigma_{\hat{\mathbf{y}}\hat{\mathbf{y}}} = \hat{H}^{(-1)}. \left(\sum_i \frac{\partial \hat{\Phi}}{\partial \vec{\mathrm{z}}_i}.\Sigma_{\mathbf{z_i z_i}}.\frac{\partial \hat{\Phi}}{\partial \vec{\mathrm{z}}_i}^{\mathrm{T}} \right).\hat{H}^{(-1)} \tag{5}$$

In our case, the data and the state are not vectors, but features in a Riemannian manifold. In fact, we can do the same derivation by replacing $(y - \hat{y})$ with $\overrightarrow{\hat{\mathrm{y}}\mathrm{y}} = J(\mathrm{f}_{\hat{\vec{\mathrm{y}}}}).(\mathrm{f}^{(-1)}_{\hat{\vec{\mathrm{y}}}} \star x)$ and $(\chi - \hat{\chi})$ with a somehow similar expression. It turns out that the definition of the covariance is changed accordingly (equation 1) and that finally nothing is changed in equation (5).

Uncertainty of the least-square mean value Using the values of the derivatives given in equation (2), we obtain:

$$\Sigma_{\hat{\mathbf{y}}\hat{\mathbf{y}}} = J(\vec{\mathrm{f}}_{\hat{\mathrm{y}}}).\left(\frac{\sum_i \Sigma_{\hat{\mathbf{z}}_i\hat{\mathbf{z}}_i}}{n^2}\right).J(\vec{\mathrm{f}}_{\hat{\mathrm{y}}})^{\mathrm{T}}$$

If all the measures are identically distributed observations of the same random feature (the least-squares estimation is optimal only in this case), we can estimate $\Sigma_{\hat{\mathbf{z}}_i\hat{\mathbf{z}}_i}$ using the residuals:

$$\Sigma_{\hat{\mathbf{y}}\hat{\mathbf{y}}} \simeq J(\vec{\mathrm{f}}_{\hat{\mathrm{y}}}).\left(\frac{\sum_i (\mathrm{f}_{\hat{\mathrm{y}}}^{(-1)} \star \vec{\mathrm{x}}_i).(\mathrm{f}_{\hat{\mathrm{y}}}^{(-1)} \star \vec{\mathrm{x}}_i)^{\mathrm{T}}}{n(n-1)}\right).J(\vec{\mathrm{f}}_{\hat{\mathrm{y}}})^{\mathrm{T}} \tag{6}$$

3.4 Example with rotations

We have plotted in figure (2) the mean and barycenter of 50 rotations randomly distributed around the "Exact" rotation (with std. dev. 60 deg, 36 deg and 18 deg along the r_x, r_y and r_z axes). Rotations are represented by their rotation vector projected on the (r_x, r_y) plane. The exterior circle represents the frontier of the definition domain ($\|r\| = \theta = \pi$). Estimations are represented with their projected uncertainty ellipsoid at $\chi^2 = 6$ in grey. The larger ellipses represent the dispersion of data around the mean value (i.e. the projected uncertainty ellipsoid of the estimated $\Sigma_{\mathbf{zz}}$ at $\chi^2 = 5$).

In the principal chart (left of figure 2) the barycenter cannot take into account the "frontier effect" (rotation vectors $r = \pi.n$ and $-r$ are identical and rotations vectors around them are close rotations). On the contrary, the mean rotation seems to be an accurate estimation. However, it is difficult to say which estimation is the best if we don't know the exact rotation. The right method to visualize the dispersion of data around the mean value is to use the exponential chart centered on this mean value (figure 2 on the right). In this chart only, the barycenter of the data corresponds to the mean value. We can see that random rotations are well grouped around the mean value (we show here the section of the rotation vector ball with the maximal dispersion, which was not the case in the principal chart).

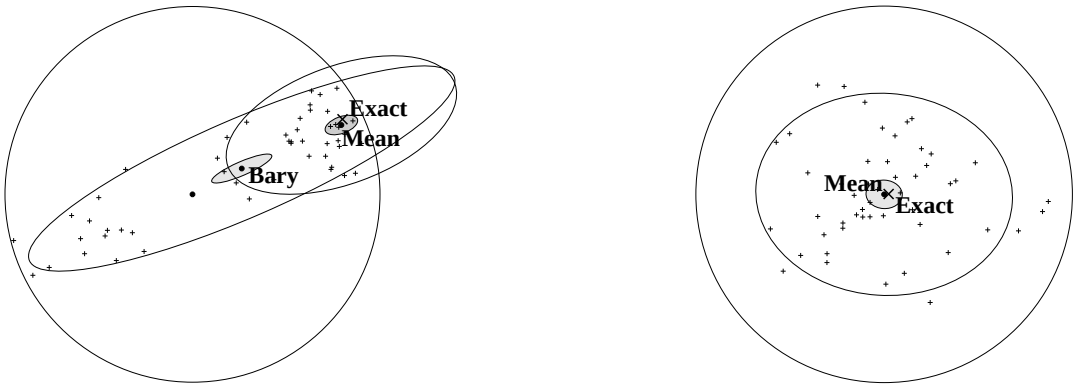

Figure 2: **Left:** Mean and barycenter of 50 random rotations around the "exact" one in the principal chart (centered around $r = 0 \sim R = Id$). **Right:** Exact and random rotations in the chart centered on the mean rotations (i.e. $\hat{\bar{r}} = 0$).

4 Fusion of uncertain features

We have developed with the least-square criterion a whole formalism to obtain the minimum of a criterion on a manifold and estimate the uncertainty of the result. We can now easily generalize this analysis to other criterions such as weighted least-squares and the Mahalanobis distance.

4.1 Weighted least-squares

Assuming that we have a confidence value p_i on each measure $\vec{\mathrm{x}}_i$, we can take it into account in the weighted least-square criterion:

$$C(\mathrm{y}) = \frac{1}{2}\sum_i p_i.\,\mathrm{dist}(\mathrm{y},\mathrm{x}_i)^2 = \frac{1}{2}\sum_i p_i.\|\vec{\mathrm{z}}_i\|^2$$

The method is rigorously similar to the least-square case, and only the derivatives are changing. We have indeed:

$$\Phi = -J(\vec{\mathrm{f}}_{\vec{\mathrm{y}}})^{(\text{-T})}\sum_i p_i.\vec{\mathrm{z}}_i \qquad H = \left(\sum_i p_i\right).J(\vec{\mathrm{f}}_{\vec{\mathrm{y}}})^{(\text{-T})}.J(\vec{\mathrm{f}}_{\vec{\mathrm{y}}})^{(\text{-1})} \qquad \frac{\partial\Phi}{\partial\vec{\mathrm{z}}_i} = -p_i.J(\vec{\mathrm{f}}_{\vec{\mathrm{y}}})^{(\text{-T})}$$

Using these new derivatives, we obtain the following gradient descent equation:

$$\vec{\mathrm{y}}_{t+1} = \vec{\mathrm{f}}_{\vec{\mathrm{y}}_t} \star \left(\frac{\sum_i p_i.\left(\vec{\mathrm{f}}^{(\text{-1})}_{\vec{\mathrm{y}}_t} \star \vec{\mathrm{x}}_i\right)}{\sum_i p_i}\right)$$

and the uncertainty at the minimum is given by

$$\Sigma_{\hat{\mathbf{y}}\hat{\mathbf{y}}} = J(\vec{\mathrm{f}}_{\hat{\mathrm{y}}}).\left(\frac{\sum_i p_i^2.\Sigma_{\hat{\mathbf{z}}_i\hat{\mathbf{z}}_i}}{(\sum p_i)^2}\right).J(\vec{\mathrm{f}}_{\hat{\mathrm{y}}})^{\mathrm{T}}$$

Usually, the weighted least-square criterion is used if we know that all residual vectors have the same (but unknown) covariance matrix Σ up the scale factors p_i: $\Sigma = p_i.\Sigma_{\vec{\mathbf{z}}_i\vec{\mathbf{z}}_i}$. In this case, Σ can be estimated from the residual vectors by $\hat{\Sigma} = \frac{1}{n-1}\sum p_i.\hat{\vec{\mathrm{z}}}_i.\hat{\vec{\mathrm{z}}}_i^{\mathrm{T}}$, and the uncertainty of the mean feature becomes:

$$\Sigma_{\hat{\mathbf{y}}\hat{\mathbf{y}}} = J(\vec{\mathrm{f}}_{\hat{\mathrm{y}}}).\left(\frac{\hat{\Sigma}}{(\sum p_i)}\right).J(\vec{\mathrm{f}}_{\hat{\mathrm{y}}})^{\mathrm{T}} = J(\vec{\mathrm{f}}_{\hat{\mathrm{y}}}).\left(\frac{\sum_i p_i.\hat{\vec{\mathrm{z}}}_i.\hat{\vec{\mathrm{z}}}_i^{\mathrm{T}}}{(n-1)(\sum p_i)}\right).J(\vec{\mathrm{f}}_{\hat{\mathrm{y}}})^{\mathrm{T}}$$

4.2 Least (Riemannian) Mahalanobis distance

To allow different and non isotropic covariance matrices for different measures, we can minimize the sum of the (squared) Mahalanobis distances with the mean value. In our framework, the Mahalanobis distance $\mu^2(\mathrm{y},\mathbf{x})$ between a fixed feature y and a random one $\mathbf{x}$ is defined in the exponential chart at the random feature.

$$\mu^2(\mathrm{y},\mathbf{x}) = \left(\mathrm{f}^{(\text{-1})}_{\vec{\mathrm{x}}} \star \vec{\mathrm{y}}\right)^{\mathrm{T}}.J(\mathrm{f}_{\vec{\mathrm{x}}})^{\mathrm{T}}.\Sigma_{\vec{\mathbf{x}}\vec{\mathbf{x}}}.J(\mathrm{f}_{\vec{\mathrm{x}}}).\left(\mathrm{f}^{(\text{-1})}_{\vec{\mathrm{y}}} \star \vec{\mathrm{x}}\right)$$

With this formulation, the Mahalanobis distance is invariant with respect to the group action: $\mu^2(\mathrm{f} \star \mathrm{y}, \mathrm{f} \star \mathbf{x}) = \mu^2(\mathrm{y}, \mathbf{x})$. We have in particular: $\mu^2(\mathrm{y}, \mathbf{x}) = \mu^2(0, \vec{\mathbf{z}})$ where $\vec{z} = \mathrm{f}_{\vec{\mathrm{y}}}^{(-1)} \star \vec{\mathrm{x}}$ is the same error vector as before and the criterion may be written

$$C(\mathrm{y}) = \frac{1}{2} \sum_i \mu^2(\mathrm{y}, \mathrm{x}_i) = \frac{1}{2} \sum_i \vec{\mathrm{z}}_i^{\mathrm{T}} . \Sigma^{(-1)}_{\mathbf{z}_i \mathbf{z}_i} . \vec{\mathrm{z}}_i$$

Thus, the algorithm is the same than for the least-squares, but the derivatives of the criterion are different. The first derivative is

$$\Phi = \left(\frac{\partial C}{\partial \vec{\mathrm{y}}} \right)^{\mathrm{T}} = \sum_i \frac{\partial \vec{\mathrm{z}}_i}{\partial \vec{\mathrm{y}}}^{\mathrm{T}} . \Sigma^{(-1)}_{\mathbf{z}_i \mathbf{z}_i} . \vec{\mathrm{z}}_i$$

Similarly, the second derivatives can be approximated by:

$$H = \frac{\partial \Phi}{\partial \vec{\mathrm{y}}} \simeq \sum_i \frac{\partial \vec{\mathrm{z}}_i}{\partial \vec{\mathrm{y}}}^{\mathrm{T}} . \Sigma^{(-1)}_{\mathbf{z}_i \mathbf{z}_i} . \frac{\partial \vec{\mathrm{z}}_i}{\partial \vec{\mathrm{y}}} \qquad \text{and} \qquad \frac{\partial \Phi}{\partial \vec{\mathrm{z}}_i} \simeq \frac{\partial \vec{\mathrm{z}}_i}{\partial \vec{\mathrm{y}}}^{\mathrm{T}} . \Sigma^{(-1)}_{\mathbf{z}_i \mathbf{z}_i}$$

Now, the Taylor expansion for the criterion is the same, and the evolution for the gradient descent is still given by equation (4). We can use the same starting value and stopping criterion as before. Practically, we have observed a convergence in about 15 iterations when starting from identity and in about 5 iterations when starting from the least-squares solution.

For the uncertainty of the solution, we replace the values of H and $\frac{\partial \Phi}{\partial \vec{z}_i}$ into equation (5) and we obtain after simplification:

$$\Sigma_{\hat{\mathbf{f}}\hat{\mathbf{f}}} = \hat{H}^{(-1)}$$

where $\hat{H}^{(-1)}$ is the Hessian matrix at the optimum.

5 Algorithm comparison

In the following, we denote by LSQ, WLSQ and MAHA the three previous methods. We analyze here the relative accuracy of the different methods, the accuracy of the uncertainty on the mean value (which should predict the absolute accuracy), and the computation times. Experiments are performed for frames, semi- or non-oriented frames, rotations and points. Frames are composed of a point and an orthonormal trihedron. In semi-oriented frames, the trihedron is composed of two directions $(t_1, t_2) \equiv (-t_1, -t_2)$ and an oriented vector n; this models the differential properties of a point on a surface. For non oriented frames, the normal is also un-oriented ($n \equiv -n$). The equations needed to implement the atomic operations are given in [PT97] for frames and rotations and in [Pen96] for the others.

Statistics are realized by randomly choosing an exact value $\vec{\mathrm{x}}$ in a given volume (uniform rotation and/or translation components). The measures $\vec{\mathrm{x}}_i$ are generated using a Gaussian noise with the requested covariance *in the exponential map at the exact value.* This means that we are using an extension of the "compositive" noise model proposed in [PT97]: the measure is $\vec{\mathrm{x}}_i = \mathrm{f}_{\vec{\mathrm{x}}} \star \vec{\mathrm{e}}_i$ where $\vec{\mathrm{e}}_i$ is a Gaussian sample with covariance $\Sigma_{\vec{\mathbf{e}}\vec{\mathbf{e}}} = J(\mathrm{f}_{\vec{\mathrm{x}}})^{(-1)} . \Sigma_{\vec{\mathbf{x}}\vec{\mathbf{x}}} . J(\mathrm{f}_{\vec{\mathrm{x}}})^{(-\mathrm{T})}$.

5.1 Relative accuracy of the methods

The absolute accuracy of a method is given by the distance between the result and the exact value (for frame based features, we compute separately the rotation and translation parts since they are in different units). As these values are varying with the set of measures and since we only want to study the relative accuracy of the three methods, we have chosen the error of MAHA as the reference and we compute the ratios $\varepsilon_{\theta_\alpha} = \theta_\alpha/\theta_{\text{MAHA}}$ and $\varepsilon_{d_\alpha} = d_\alpha/d_{\text{MAHA}}$ to characterize the relative accuracy of $\alpha = \text{LSQ}$ and $\alpha = \text{WLSQ}$. This ratio indicates a better accuracy of method α with respect to MAHA if $\varepsilon < 1$ and conversely. Since the three algorithms are related to three different hypotheses on the noise on features, we investigate each type of noise distribution separately.

Independent and Identically Distributed measurements In the case of IID measurements, error vectors share the same covariance matrix: all weights are 1 and WLSQ is strictly identical to LSQ. There is no significant difference either between the results of LSQ and MAHA. This was expected since the characterization of the MAHA and LSQ optimums corresponds if $\Sigma_{\vec{z}_i\vec{z}_i} = \Sigma$:

$$\sum_i \Sigma_{\vec{z}_i\vec{z}_i}.\vec{z}_i = 0 \quad \Leftrightarrow \quad \Sigma.\sum_i \vec{z}_i = 0 \quad \Leftrightarrow \quad \sum_i \vec{z}_i = 0$$

Independent and Similarly Distributed measurements In the case of ISD measurements, error vectors have the same covariance matrix up to a known scale factor. We have chosen to sample feature i according to the covariance $\Sigma_{\vec{e}_i\vec{e}_i} = \frac{1}{i}.\Sigma$. Thus, the weights are $p_i = 1/i$. The matrix Σ is taken to be diagonal with standard deviations (1, 5, 10) for points and (0.3, 0.6, 0.9) in radian for rotations or trihedrons.

To obtain reliable statistics, we have done 250 trials for each number of features and we present in figure (3) the mean value and the standard deviation[3] of the relative accuracies of LSQ and WLSQ with respect to MAHA.

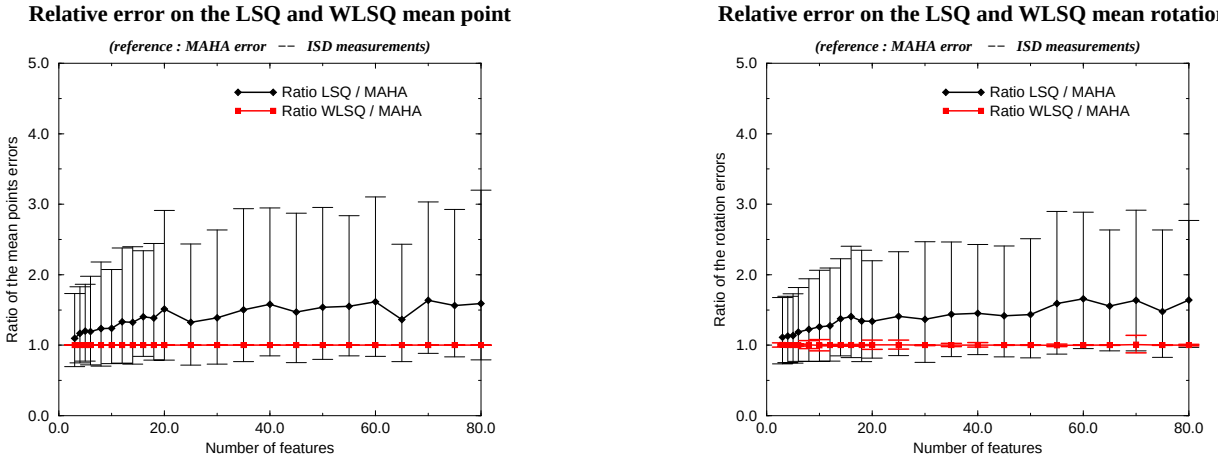

Figure 3: Relative accuracy of the mean point (on the left) and of the mean rotations (on the right) estimated by LSQ and WLSQ with respect to MAHA, in the case of ISD measurements.

[3]As the relative accuracy is multiplicative (2 times more accurate means $\varepsilon = 0.5$), we have computed the mean value and the standard deviation of $\log(\varepsilon)$, but we present the result with ratios for an easier visualization.

We can observe that there is still no significant difference between the results of WLSQ and MAHA, but LSQ can be much less precise (1.5 less precise in the mean for our example). The comparison of the figure (3 left) and (3 right) stresses the very similar behavior of the algorithms for points and rotations. In fact we also obtain very similar results for the other kinds of features.

Independent but Differently Distributed measurements To simulate the case where all measurements have very different (but known) noises, we use for each measurement the diagonal covariance of IID measurements where each term is multiplied by an independent random sample λ uniformly distributed between 0 and 1. For WLSQ, we have used the weights $p_i = \det(\Sigma_{\vec{\mathbf{e}}_i \vec{\mathbf{e}}_i})^{-1/k}$ where k is the dimension of the manifold (3 for points and rotations, 6 for frame-based features).

This time, there is no significant difference between the relative accuracy of LSQ and WLSQ, but they are both much less precise than MAHA. In this experiment, we observed a relative inaccuracy of a factor 1.5 to 2. In other experiments, we took $\lambda = \exp(-5.\mu^2)$ with μ uniform between 0 and 1, and LSQ and WLSQ were much less accurate. In this last case, the covariance matrices on features do have more different "sizes" and WLSQ performs 5% to 10% better than LSQ.

5.2 Accuracy of the uncertainty estimation

The validation index With synthetic data, we know the exact feature that we are looking for, and we can verify that it is compatible with our estimation modulo its uncertainty. The basic idea of the validation method presented in [PT97] is to normalize the difference between the expected and the estimated feature with respect to the predicted uncertainty. In fact, the simplest way to do that is to compute the Mahalanobis distance between the computed mean $\hat{\mathrm{y}}$ and the exact feature y. Under the Gaussian hypothesis (justified in [Pen96]), $\mu^2(\mathrm{y}, \hat{\mathbf{y}})$ should be χ^2_k distributed if the covariance matrix is exact (k is the dimension of the manifold).

We repeat this experiment on N series of measurements to obtain N independent values μ_i^2. The Kolmogorov-Smirnov (or simply K-S) test [PFTV91] is well adapted to verify if it is really χ^2_k distributed, but since it only gives a binary answer, we also use the fact that the mean value of a χ^2_k distribution is k and its variance is $2.k$. We call *validation index* $I = \frac{1}{N}\sum \mu_i^2$ the estimated mean value of μ_i^2. This index can be interpreted as an indication on how the estimation method under-estimates ($I > k$) or over-estimates ($I < k$) the covariance matrix of the result. It is a kind of relative error on the error estimation.

This method can be generalized to the case of real data by splitting the set of measurements in two approximatively equal sets to obtain two independent estimates $\mathbf{y}_1$ and $\mathbf{y}_2$ of the same feature y. Their Mahalanobis distance $\mu^2(\mathbf{y}_1, \mathbf{y}_2)$ should once again be χ^2_k distributed under the Gaussian assumption. The two estimated features could then be merged together (using MAHA or any other algorithm) to obtain a better estimation of the transformation and used for other purposes.

Known covariance on features In fact, it turns out that all three algorithm give a very good prediction of their accuracy when the covariance on individual features is known (and used in the computations...).

We present in figure (4 left) the computed mean and standard deviation of the validation index with LSQ on rotations in the case of IDD measurements. The dotted lines represent the

theoretical values for a χ^2_3 distribution (a mean of 3 and a standard deviation of $\sqrt{6} \simeq 2.45$). The K-S test at 1% is always accepted. The mean validation index (over 6000 estimations) is ($I = 3.01$, $\sigma_I^2 = 6.18$) for LSQ, ($I = 2.99$, $\sigma_I^2 = 5.85$) for WLSQ, and ($I = 2.99$, $\sigma_I^2 = 6.08$) for MAHA, which proves that the uncertainty estimation is very accurate. This indicates that over- or under-estimation of the covariance matrix are less than 1%. The same results are observed with other kind of features.

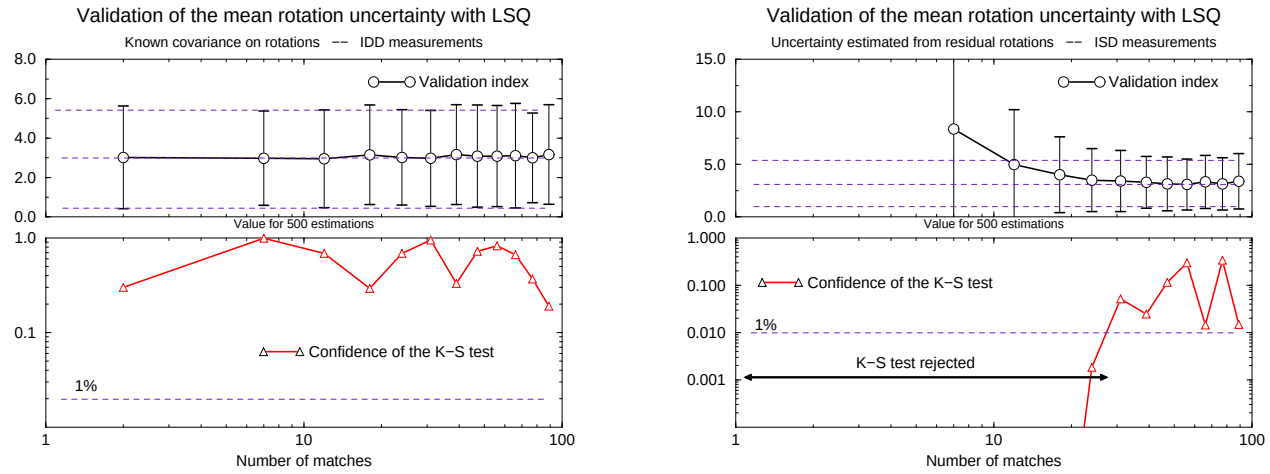

Figure 4: Validation of the mean uncertainty prediction with a known covariance on rotations with LSQ (**left**) and the noise on data estimated from the residuals (**right**). **Top:** mean and standard deviation of the validation index with respect to the number of matches used. **Bottom:** confidence value of the K-S test. It validates the uncertainty with a confidence superior to 1% in all cases when the covariance on features is known, but only for the mean of more than 30 features if the noise is estimated from the residuals.

Estimating the noise from residuals With LSQ, we can compute the uncertainty of the mean feature even if we have no information about the noise on features by assuming IID measurements and estimating the covariance from residuals. However, this process is much more error-prone than the computation of the mean itself. In the case of figure (4 right), we need a minimum number of 30 rotations to validate our uncertainty estimation. What is interesting and more surprising is that we have obtained the same results for all types of data noise (IID, ISD and IDD). The dimension of the estimated mean seems to be the only factor that influences the minimum of features: we have still the same results for points, but we need from 50 to 60 features to validate the uncertainty estimation for frame-based features.

For WLSQ, we still need a minimum of 30 rotations or points to obtain a validated uncertainty with ISD measurements, but the uncertainty remains underestimated by 20% to 30% for IDD measurements, even with more measurements. Hence, WLSQ seems to be interesting only in the ISD measurements hypothesis.

5.3 Computation times

We observed that the computation time for a given algorithm with a given type of features is linear in the number of feature, which could be guessed from the algorithms equations. Thus, we can focus on a fixed number of measurements (table 1). We have obtained similar results either

with IID, ISD or IDD measurements and with the different uncertainty estimation schemes for LSQ and WLSQ.

	Frames	Semi-oriented Frames	Non-oriented Frames	Rotations	Points
LSQ	28.6 ms	30.3 ms	33.0 ms	16.8 ms	6.0 ms
WLSQ	29.7 ms	31.0 ms	32.4 ms	17.8 ms	6.2 ms
MAHA	266.6 ms	272.4 ms	277.4 ms	65.1 ms	15.4 ms

Table 1: Computation times for computing the mean value of 100 features on an ultra-sparc 143 Mhz running Solaris 2.5. Reported values are user time values averaged on 1000 trials.

As expected, MAHA is the slowest algorithm: it is 3 to 8 times longer than LSQ and WLSQ for the same type of features. In this experiment, all three algorithms were initialized with the first measure of the series. We note that initializing MAHA with the result of LSQ makes it converge a little bit faster but the extra cost of LSQ makes the total time almost identical. It is interesting to note that computing the mean rotation or the mean frame with LSQ (or WLSQ) takes only 3 to 5 times longer than computing the mean point and its uncertainty[4].

Thus, we should prefer LSQ and WLSQ for IID and ISD measurements since they are accurate and faster in these cases. For IDD measurements, there is a tradeoff between the accuracy and the computation time. For real time applications, or when the estimation is often repeated, LSQ and WLSQ are probably sufficient. If we want a precise measurement, MAHA is better but takes longer.

6 Conclusion

We present in this article a detailed study of how to minimize a criterion on a Riemannian manifold, and how to compute the uncertainty of this estimation. This framework is particularly well adapted to the computation of the mean of a set of features. We have investigated three criterions: the least-squares (LSQ), the weighted least-squares (WLSQ) and the minimization of the Mahalanobis Distance (MAHA). For each criterion, we show that our intrinsic gradient descent algorithm applies very simply and that the uncertainty on the resulting estimation can be easily computed from the covariance matrices of the measurements.

Experiments show that the three algorithms perform very well in all conditions, even if MAHA is more accurate than the other in the case of measurements with different covariance matrices. They all predict the accuracy of their result within 1% (in the mean), which is quite remarkable. In the case where the uncertainty on measurements is unknown, one can estimate the noise from residual vectors for LSQ and WLSQ. Both methods need a minimum number of features to be able to predict accurately their uncertainty. LSQ predicts surprisingly well its accuracy even if measurements have very different uncertainties. It is worth notice that computing the mean rotation and its uncertainty takes only 3 to 4 times longer than for points. This emphasizes the relative efficiency of our algorithms.

One of the most striking result in these experiments is the very strong similarity of the algorithms behavior for all types of features with respect to points. This stresses the validity

[4]Computation times for points are given here for the application of our generic implementation to points. The direct computation of the barycenter and its uncertainty takes between 0.5 and 1 ms for 100 points.

of our approach using Riemannian geometry and lets us anticipate that other statistical results and algorithms could be generalized to manifolds in this framework.

References

[Alt86] S. L. Altmann. *Rotations, Quaternions, and Double Groups.* Clarendon Press - Oxford, 1986.

[Car92] M. do Carmo. *Riemannian Geometry.* Mathematics. Birkhäuser, Boston, Basel, Berlin, 1992.

[Den96] B. S. Denney. *Rotation Clustering for Robot Vision.* PhD thesis, University of California, Irvine, 1996.

[FA96] J. Feldmar and N. Ayache. Rigid, affine and locally affine registration of free-form surfaces. *IJCV*, 18(2):99–119, May 1996.

[GMW81] P.E. Gill, W. Murray, and M.H. Wright. *Practical optimization.* Academic Press, London and New York, 1981.

[Gri90] W.E.L. Grimson. *Object Recognition by Computer - The role of Geometric Constraints.* MIT Press, 1990.

[Kli82] W. Klingenberg. *Riemannian Geometry.* Walter de Gruyter, Berlin, New York, 1982.

[PA97] X. Pennec and N. Ayache. Uniform distribution, distance and expectation problems for geometric features processing. *Journal of Mathematical Imaging and Vision*, in press, 1997. Also as INRIA research report n^o 2820.

[Pen96] Xavier Pennec. *L'Incertitude dans les Problèmes de Reconnaissance et de Recalage – Applications en Imagerie Médicale et Biologie Moléculaire.* PhD thesis, Ecole Polytechnique, Palaiseau (France), december 1996.

[PFTV91] W.H. Press, B.P. Flannery, S.A Teukolsky, and W.T. Vetterling. *Numerical Recipices in C.* Cambridge Univ. Press, 1991.

[PT97] X. Pennec and J.P. Thirion. A framework for uncertainty and validation of 3D registration methods based on points and frames. *Int. Journal of Computer Vision*, 25(3):203–229, 1997.

[Spi79] M. Spivak. *Differential Geometry*, volume 1. Publish or Perish, Inc., 2nd edition, 1979.

[ZRS92] H. Zhuang, Z.S. Roth, and R. Sudhakar. Practical fusion algorithms for rotation matrices: A comparative study. *J. of Robotic Systems*, 9(7):915–931, 1992.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - B.P. 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Lorraine : Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 Villers lès Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot St Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, B.P. 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399